

студенты чувствуют себя более комфортно, находясь в равных условиях, так как способны прочесть текст, осмыслить его и дать верный ответ.

Тестовые задания позволяют оценить уровень овладения теоретическим материалом и навыки каждого студента. Выполнение данных заданий помогает студентам самостоятельно совершенствоваться и углублять знания по химии, способствует усилению интереса к данному предмету, позволяет использовать полученные знания в повседневной жизни.

Бабаев С.К.

КОМПЬЮТЕРНЫЕ ТЕСТЫ КАК СРЕДСТВО КОНТРОЛЯ ЗНАНИЙ СТУДЕНТОВ

vbroad@narod.ru

КФ НГПУ

г. Куйбышев Новосибирской области

Среди средств диагностики знаний наиболее эффективными являются тесты. Трудности при создании тестов могут составить: подготовка корректных заданий, а также выработка объективной системы оценок. Всё чаще в современном мире проведение таких тестов и их оценка производится при помощи ЭВМ. На данный момент существует множество интегрированных систем разработок. Но для создания наиболее простой и соответственно наиболее понятной для обычного пользователя программы можно использовать Visual Basic. Преимуществами VB являются: лёгкий синтаксис и интуитивно понятный интерфейс пользователя. Что позволяет быстро изучить основные концепции программирования в данной среде разработки. Наличие в Visual Basic библиотеки компонентов представляет разработчику возможность создать наиболее, по его понятию, удобную

навигацию по тест – программе, а также сделать более наглядным результат, полученный студентом при прохождении тестирования.

Нами была разработана тест – программа по курсу дифференциальных уравнений. Перед прохождением непосредственно самого теста каждый пользователь заполняет прототип анкеты. В анкете присутствуют следующие поля: фамилия, имя, отчество, учебное учреждение, факультет, группа. Основой для размещения различных элементов управления и самих данных служит форма (Form). Для реализации анкеты использовались такие элементы Visual Basic как: label (надпись), Text (текстовое поле), CommandButton (кнопки управления). Далее приведена таблица элементов и заданных для них свойств Caption или Text в зависимости от вида элемента.

Элемент	Свойства	
	Caption	Text
Label1	Фамилия	-
Label2	Имя	-
Label3	Отчество	-
Label4	Учебное заведение	-
Label5	Факультет	-
Label6	Курс	-
Label7	Группа	-
Text 1,2,3,4,5,6,7	-	“ ”
CommandButton1	&Выход	-
CommandButton2	&Принять	-
CommandButton3	&Далее	-

После нажатия на кнопку “Принять” введённые пользователем данные заносятся в соответствующий файл в рабочем каталоге программы. Эти же сведения заносятся в итоговую форму для вывода результатов, где

из данных введенных студентом сформируется обращение с объявлением результатов. Производится это при помощи конструкции

`Form_a.Label_i.Caption= Form_b.Text_c.Text .`

Где a - порядковый номер результирующей формы, i – порядковый номер соответствующей надписи на форму, b – порядковый номер формы-анкеты и c – порядковый номер текстового поля, сведения с которого мы хотим передать на Label в результирующую форму.

К данным, занесённым в файл, позднее добавится также и оценка пользователя, что позволит использовать данный файл для обработки статистическими программами, в том числе спроектированными, на Visual Basic, специфически для абстрагирования результатов проведения тестов.

Более детально рассмотрим способ представления самого материала. Сам материал разбит на две части: теоретическую и практическую. В теоретической части, как и следует из названия, студенту предлагают выбрать верную формулировку определения или теоремы, а в практической проверяются навыки решения, т.е. студенту даётся задание и предлагаются четыре варианта ответа. Для предотвращения случайного выбора неверного ответа использовался элемент Option, который располагался сразу возле варианта ответа. Для фиксации нажатия на элемент Option обрабатываем его событие `click()`, отвечающее за одинарное нажатие левой клавишей мыши. Каждый ответ имеет своё количество баллов, присваиваемое студенту за выбор этого варианта ответа. Если ответ совершенно не верен, то количество баллов за него конечно равно нулю. На протяжении всего тестирования эти баллы суммируются и на основе процентного выполнения работы выставляется оценка. Например, присвоение того или иного количества баллов осуществляется следующим образом

```

Private Sub Option1_Click()
    Dim I As Byte
    I = Val(Form_а.lblOt.Caption)
    I=I+15
    Form_а.lblMak.Caption = I
End Sub

```

Где *Option1* – переключатель, соответствующий некоторому варианту ответа, при выборе которого студент получает 15 баллов. Строка *Form_а.lblOt.Caption* указывает, что осуществляется ссылка на надпись *lblOt* результирующей формы с порядковым номером *а*. Для реализации самого присваивания результата объявляем некоторую переменную *I* типа Byte. При помощи Val преобразуем значение *Form_а.lblMak.Caption* в числовое представление, прибавляем необходимое количество баллов и возвращаем это значение надписи.

После выбора одного из вариантов все элементы формы, кроме кнопки Далее (позволяющей студенту перейти к следующему вопросу), блокируются. Добиться этого эффекта можно путём изменения значения свойства Enabled, присутствующего практически у всех задействованных нами элементов. Наглядно это можно продемонстрировать следующим фрагментом:

```

Private Sub Option1_Click()
    Option4.Enabled = False
    Option3.Enabled = False
    Option2.Enabled = False
    Option1.Enabled = False
End Sub

```

Как видно из этого примера, при выборе одного варианта возможность повторного выбора блокируется.

В случае если студент решил не отвечать на вопрос, а сразу воспользоваться кнопкой *Далее* ответ на этот вопрос не будет ему зачтён, т.е. количество баллов от пропуска вопроса не увеличится, что скажется на статистике процентного выполнения теста.

Если потенциальная тестирующая программа не подразумевает наличия большого количества вопросов, то их заранее можно разместить на определённом количестве форм и обеспечить безотказную систему перехода, пример этого и приведён ниже:

```
Private Sub cmdNext_Click()  
    Me.Hide  
    Form2.Show vbModal  
End Sub
```

Строка *Me.Hide* позволит скрыть форму, с которой осуществляется переход, а *Form2.Show vbModal* – отобразить следующую форму с вопросом (в нашем случае вторую).

Если же предполагается, что в программе будет находиться база вопросов, которая в случайном порядке будет предлагать вопросы студенту (что и реализовано в опытном образце), то для этого можно воспользоваться генератором случайных чисел, поставив в соответствие каждому возможному варианту определённый вопрос:

```

Private Sub Command3_Click()\
    Dim I As Byte
    I = Int((2*Rnd)+1)
    Select Case i
        Case is = 1
            Form17.Show vbModal
        Case is = 2
            Form18.Show vbModal
    End Select
End Sub

```

Строкой $I = \text{Int}((2 * \text{Rnd}) + 1)$ мы задаём диапазон возможных значений переменной I и ассоциируем с ними определённые вопросы теста.

Заключительным этапом в создании тест – программ является подсчёт результатов. Как уже упоминалось, он делается на основе присвоенных баллов, которые, в сущности, отображают процентное выполнение работы. На основе этих результатов и выставляется более привычная оценка по пятибалльной системе.

Из главного меню, которое становится доступным после прохождения теста, существует возможность перейти снова к анкете или к выбору стиля теста, что позволяет сразу пройти весь тестовый комплекс, получив сведения о своей теоретической и практической подготовке.